On Problems as Hard as CNF-SAT[1]

Marek Cygan, University of Warsaw, Poland. cygan@mimuw.edu.pl.
Holger Dell, Saarland University and Cluster of Excellence (MMCI), Germany. hdell@mmci.uni-saarland.de.
Daniel Lokshtanov, University of Bergen, Norway. daniello@ii.uib.no.
Dániel Marx, Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI), Budapest, Hungary. dmarx@cs.bme.hu.
Jesper Nederlof, Technische Universiteie Eindhoven, The Netherlands. j.nederlof@tue.nl.
Yoshio Okamoto, University of Electro-Communications, Japan. okamotoy@uec.ac.jp.
Ramamohan Paturi, University of California, San Diego, USA. paturi@cs.ucsd.edu.
Saket Saurabh, Institute of Mathematical Sciences, India. saket@imsc.res.in.
Magnus Wahlström, Royal Holloway, University of London, UK. Magnus.Wahlstrom@rhul.ac.uk.

The field of exact exponential time algorithms for NP-hard problems has thrived over the last decade. While exhaustive search remains asymptotically the fastest known algorithm for some basic problems, non-trivial exponential time algorithms have been found for a myriad of problems, including Graph Coloring, Hamiltonian Path, Dominating Set and 3-CNF-Sat. In some instances, improving these algorithms further seems to be out of reach. The CNF-Sat problem is the canonical example of a problem for which the trivial exhaustive search algorithm runs in time $O(2^n)$, where n is the number of variables in the input formula. While there exist non-trivial algorithms for CNF-Sat that run in time $o(2^n)$, no algorithm was able to improve the *growth rate* 2 to a smaller constant, and hence it is natural to conjecture that 2 is the optimal growth rate. The *strong exponential time hypothesis* (SETH) by Impagliazzo and Paturi [JCSS 2001] goes a little bit further and asserts that, for every $\epsilon < 1$, there is a (large) integer k such that k-CNF-Sat cannot be computed in time $2^{\epsilon n}$.

In this paper, we show that, for every $\epsilon < 1$, the problems Hitting Set, Set Splitting, and NAE-Sat cannot be computed in time $O(2^{\epsilon n})$ unless SETH fails. Here n is the number of elements or variables in the input. For these problems, we actually get an equivalence to SETH in a certain sense. We conjecture that SETH implies a similar statement for Set Cover, and prove that, under this assumption, the fastest known algorithms for Steiner Tree, Connected Vertex Cover, Set Partitioning, and the pseudo-polynomial time algorithm for Subset Sum cannot be significantly improved. Finally, we justify our assumption about the hardness of Set Cover by showing that the parity of the number of solutions to Set Cover cannot be computed in time $O(2^{\epsilon n})$ for any $\epsilon < 1$ unless SETH fails.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Computations on discrete structures*

[1]An extended abstract of this paper appeared in the proceedings of CCC 2012.

M.C. was partially supported by National Science Centre grant no. N206 567140, Foundation for Polish Science and ONR Young Investigator award when at the University at Maryland. H.D.'s research was partially supported by the Alexander von Humboldt Foundation and NSF grant 1017597. D.M.'s research was supported by ERC Starting Grant PARAMTIGHT (280152). J.N. was supported by NWO project "Space and Time Efficient Structural Improvements of Dynamic Programming Algorithms." Y.O. was partially supported by Grant-in-Aid for Scientific Research from Japan Society for the Promotion of Science. R.P.'s research was supported by NSF grants CCF-1213151 and CCF-0947262 from the Division of Computing and Communication Foundations. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

DOI: http://dx.doi.org/10.1145/0000000.0000000

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Optimal growth rate, Reduction, Satisfiability, Stong exponential time hypothesis

ACM Reference Format:
ACM Trans. Algor. V, N, Article A (January YYYY), 24 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Every problem in NP can be solved in time $2^{\mathrm{poly}(m)}$ by brute force, that is, by enumerating all candidates for an NP-witness, which is guaranteed to have length polynomial in the input size m. While we do not believe that polynomial time algorithms for NP-complete problems exist, many NP-complete problems have exponential time algorithms that are dramatically faster than the naïve brute force algorithm. For some classical problems, such as SUBSET SUM or HAMILTONIAN CYCLE, such algorithms were known [Held and Karp 1962; Bellman 1962] even before the concept of NP-completeness was discovered. Over the last decade, a subfield of algorithms devoted to developing faster exponential time algorithms for NP-hard problems has emerged. A myriad of problems have been shown to be solvable much faster than by naïve brute force, and a variety of algorithm design techniques for exponential time algorithms has been developed.

What the field of exponential time algorithms sorely lacks is a complexity-theoretic framework for showing running time lower bounds. Some problems, such as INDEPENDENT SET and DOMINATING SET have seen a chain of improvements [Fomin et al. 2009; van Rooij et al. 2009; Robson 1986; Kneis et al. 2009], each new improvement being smaller than the previous. For these problems, the running time of the discovered algorithms seems to converge towards $O(C^n)$ for some unknown constant C, where n denotes the number of vertices of the input graphs. For other problems, such as GRAPH COLORING or STEINER TREE, non-trivial algorithms have been found, but improving the *growth rate* C of the running time any further seems to be out of reach [Björklund et al. 2009; Nederlof 2009]. The purpose of this paper is to develop tools that allow us to explain why we are stuck for these problems. Ideally, for any problem whose best known algorithm runs in time $O(C^n)$, we want to prove that the existence of $O(c^n)$-time algorithms for any constant $c < C$ would have implausible complexity-theoretic consequences.

Previous Work. Impagliazzo and Paturi's *Exponential Time Hypothesis* (ETH) addresses the question whether NP-hard problems can have algorithms that run in "subexponential time" [Impagliazzo and Paturi 2001]. More precisely, the hypothesis asserts that 3-CNF-SAT cannot be computed in time $2^{o(n)}$, where n is the number of variables in the input formula. ETH is considered to be a plausible complexity-theoretic assumption, and subexponential time algorithms have been ruled out under ETH for many decision problems [Impagliazzo et al. 2001], parameterized problems [Chen et al. 2005; Lokshtanov et al. 2011], approximation problems [Marx 2007], and counting problems [Dell et al. 2012]. However, ETH does not seem to be sufficient for pinning down what exactly the best possible growth rate is. For this reason, we base our results on a stronger hypothesis.

The fastest known algorithms for CNF-SAT have running times of the form $2^{n-o(n)}\mathrm{poly}(m)$ [Schuler 2005; Williams 2011], which does not improve upon the growth rate 2 of the naïve brute force algorithm that runs in time $2^n\mathrm{poly}(m)$. Hence a natural candidate for a stronger hypothesis is that CNF-SAT cannot be computed in time $2^{\epsilon n}\mathrm{poly}(m)$ for any $\epsilon < 1$. However, we do not know whether our lower bounds on the growth rate of specific problems can be based on this hypothesis. The main techni-

cal obstacle is that we have no analogue of the sparsification lemma, which applies to k-CNF formulas and makes ETH a robust hypothesis [Impagliazzo et al. 2001]. In fact, very recent results indicate that such a sparsification may be impossible for general CNF formulas [Santhanam and Srinivasan 2011]. For this reason, we consider the *Strong Exponential Time Hypothesis* (SETH) of Impagliazzo and Paturi [Impagliazzo and Paturi 2001; Impagliazzo et al. 2001; Calabro et al. 2009]. This hypothesis asserts that, for every $\epsilon < 1$, there is a (large) integer k such that k-CNF-SAT cannot be computed by any bounded-error randomized algorithm in time $O(2^{\epsilon n})$. In particular, SETH implies the hypothesis for CNF-SAT above, but we do not know whether they are equivalent. Since SETH is a statement about k-CNF formulas for constant $k = k(\epsilon)$, we can apply the sparsification lemma for every fixed k, which allows us to use SETH as a starting point in our reductions.

Our results. Our first theorem is that SETH is equivalent to lower bounds on the time complexity of a number of standard NP-complete problems.

THEOREM 1.1. *Each of the following statements is equivalent to SETH.*

(*1*) *For all $\epsilon < 1$, there exists k such that k-CNF-SAT, the satisfiability problem for n-variable k-CNF formulas, cannot be solved in time $O(2^{\epsilon n})$.*
(*2*) *For all $\epsilon < 1$, there exists k such that k-HITTING SET, the hitting set problem for set systems over $[n]$ with sets of size at most k, cannot be solved in time $O(2^{\epsilon n})$.*
(*3*) *For all $\epsilon < 1$, there exists k such that k-SET SPLITTING, the set splitting problem for set systems over $[n]$ with sets of size at most k, cannot be solved in time $O(2^{\epsilon n})$.*
(*4*) *For all $\epsilon < 1$, there exists k such that k-NAE-SAT, the not-all-equal satisfiability problem for n-variable k-CNF formulas, cannot be solved in time $O(2^{\epsilon n})$.*
(*5*) *For all $\epsilon < 1$, there exists c such that c-VSP-CIRCUIT-SAT, the satisfiability problem for n-variable series-parallel circuits of size at most cn, cannot be solved in time $O(2^{\epsilon n})$.*

For all of the above problems, the naïve brute force algorithm runs in time $O(2^n)$. While there may not be a consensus that SETH is a "plausible" complexity-theoretic assumption, our theorem does indicate that finding an algorithm for CNF-SAT whose growth rate is smaller than 2 is as difficult as finding such an algorithm for any of the above problems. Since our results are established via suitable reductions, this can be seen as a completeness result under these reductions. Moreover, we actually prove that the optimal growth rates for all of the problems above are *equal* as k tends to infinity. This gives an additional motivation to study the Strong Exponential Time Hypothesis.

An immediate consequence of Theorem 1.1 is that, if SETH holds, then CNF-SAT, HITTING SET, SET SPLITTING, NAE-SAT, and the satisfiability problem of series-parallel circuits do not have bounded-error randomized algorithms that run in time $2^{\epsilon n}\mathrm{poly}(m)$ for any $\epsilon < 1$. All of these problems are *search* problems, where the objective is to find a particular object in a search space of size 2^n. Of course, we would also like to show tight connections between SETH and the optimal growth rates of problems that *do* have non-trivial exact algorithms. Our prototypical such problem is SET COVER: Given a set system with n elements and m sets, we want to select a given number t of sets that cover all elements. Exhaustively trying all possible ways to cover the elements takes time at most $2^m\mathrm{poly}(m)$. However, m could be much larger than n, and it is natural to ask for the best running time that one can achieve in terms of n. It turns out that a simple dynamic programming algorithm [Fomin et al. 2004] can solve SET COVER in time $2^n\mathrm{poly}(m)$. The natural question is whether the growth rate of this simple algorithm can be improved. While we are not able to resolve this question, we connect the existence of an improved algorithm for SET COVER to the existence of faster algorithms for several problems. Specifically, we show the following theorem.

THEOREM 1.2. *Assume that, for all $\epsilon < 1$, there exists k such that k-SET COVER, the set cover problem for set systems over $[n]$ with m sets of size at most k, cannot be solved in time $2^{\epsilon n}\mathrm{poly}(m)$. Then, for all $\epsilon < 1$, we have the following.*

(*1*) STEINER TREE *cannot be solved in time $2^{\epsilon t}\mathrm{poly}(n)$, where n is the number of vertices and t is the size of a solution,*
(*2*) CONNECTED VERTEX COVER *cannot be solved in time $2^{\epsilon t}\mathrm{poly}(n)$, where n is the number of vertices and t is the size of a solution,*
(*3*) SET PARTITIONING *cannot be solved in time $2^{\epsilon n}\mathrm{poly}(m)$, where n is the size of the universe and m is the number of hyperedges, and*
(*4*) SUBSET SUM *cannot be solved in time $t^{\epsilon}\mathrm{poly}(n)$, where n is the size of the universe and t is a target integer.*

All problems mentioned in this theorem have non-trivial algorithms whose running times are as above with $\epsilon = 1$ [Björklund et al. 2007; Nederlof 2009; Cygan et al. 2011; Fomin et al. 2004; Cormen et al. 2009]. Under the assumption in the theorem, we therefore obtain tight lower bounds on the growth rate of exact algorithms for STEINER TREE, CONNECTED VERTEX COVER, SET PARTITIONING, and SUBSET SUM. The best currently known algorithms for these problems share two interesting common features. First, they are all *dynamic programming* algorithms. Thus, Theorem 1.2 hints at SET COVER being a "canonical" dynamic programming problem. Second, the algorithms can all be modified to compute the number of solutions modulo two in the same running time. In fact, the currently fastest algorithm [Cygan et al. 2011] for CONNECTED VERTEX COVER works by reducing the problem to computing the number of solutions modulo two.

While Theorem 1.1 is an equivalence, Theorem 1.2 is not. One might ask whether it is possible to find reductions back to SET COVER and to strengthen Theorem 1.2 in this manner. We believe that this would be quite difficult: A suitable reduction from, say, STEINER TREE to SET COVER that proves the converse of Theorem 1.2 would probably also work for $\epsilon = 1$. This would give an alternative proof that STEINER TREE can be computed in time $2^t\mathrm{poly}(m)$. Hence, finding such a reduction is likely to be a challenge since the fastest known algorithms [Björklund et al. 2007; Nederlof 2009] for STEINER TREE are quite non-trivial — it took more than 30 years before the classical $3^t\mathrm{poly}(n)$-time Dreyfus–Wagner algorithm for STEINER TREE was improved to $2^t\mathrm{poly}(n)$. Similar comments apply to CONNECTED VERTEX COVER since its $2^t\mathrm{poly}(n)$-time algorithm is quite complex [Cygan et al. 2011].

The hardness assumption for SET COVER in Theorem 1.2 needs some justification. Ideally we would like to replace this assumption with SETH, that is, we would like to prove that SETH implies the hardness assumption for SET COVER in Theorem 1.2. We do not know a suitable reduction, but we are able to provide a different kind of evidence for hardness: We show that a $2^{\epsilon n}\mathrm{poly}(m)$-time algorithm to compute the number of set covers modulo two would violate $\oplus$-SETH, which is a hypothesis that implies SETH. Formally, $\oplus$-SETH asserts that, for all $\epsilon < 1$, there exists a (large) integer k such that k-CNF-$\oplus$SAT cannot be computed in time $O(2^{\epsilon n})$. Here, k-CNF-$\oplus$SAT is the problem of computing the number of satisfying assignments of a given k-CNF formula modulo two. It follows from known results [Calabro et al. 2003; Traxler 2008] (see also Section 3.1) that, if SETH holds, then so does $\oplus$-SETH. As a partial justification for the hardness assumption for SET COVER in Theorem 1.2, we provide the following theorem.

THEOREM 1.3.

(*1*) *For all $\epsilon < 1$, there exists k such that k-CNF-$\oplus$SAT, the parity satisfiability problem for n-variable k-CNF formulas, cannot be solved in time $O(2^{\epsilon n})$.*

(2) *For all* $\epsilon < 1$*, there exists* k *such that* k-$\oplus$ALL HITTING SETS*, the parity hitting set problem for set systems over* $[n]$ *with sets of size at most* k*, cannot be solved in time* $O(2^{\epsilon n})$.
(3) *For all* $\epsilon < 1$*, there exists* k *such that* k-$\oplus$ALL SET COVERS*, the parity set cover problem for set systems over* $[n]$ *with sets of size at most* k*, cannot be solved in time* $O(2^{\epsilon n})$.

In the statement of Theorem 1.3, the $\oplus$ALL HITTING SETS and $\oplus$ALL SET COVERS problems are defined as follows: the input is a set system and the objective is to compute the parity of the number of hitting sets (resp. set covers) in the system. An immediate consequence of Theorem 1.3 that we find interesting is that $\oplus$-SETH rules out the existence of $2^{\epsilon n}\mathrm{poly}(m)$-time algorithms to compute the number of set covers of a set system, for any $\epsilon < 1$.

Theorem 1.3 together with the fact that the algorithms for all problems mentioned in Theorem 1.2 can be modified to count solutions modulo two leads to the following questions: Can we show running time lower bounds for the counting versions of these problems? We show that this is indeed possible. In particular we show that, assuming $\oplus$-SETH, there is no $2^{\epsilon t}\mathrm{poly}(n)$-time algorithm that computes the parity of the number of Steiner trees that have size at most t, and no $2^{\epsilon t}\mathrm{poly}(n)$-time algorithm that computes the parity of the number of connected vertex covers that have size at most t. Thus, unless $\oplus$-SETH fails, any improved algorithm for SET COVER, STEINER TREE, or CONNECTED VERTEX COVER cannot be used to compute the parity of the number of solutions.

We find it intriguing that SETH and $\oplus$-SETH can be used to show tight running time lower bounds, sometimes for problems for which the best algorithm has been improved several times, such as for STEINER TREE or CONNECTED VERTEX COVER. We feel that such sharp bounds are unlikely to just be a coincidence, leading us to conjecture that the relationship between the considered problems is even closer than what we show. Specifically, we conjecture that SETH implies the hardness assumption for SET COVER in Theorem 1.2. This conjecture provides an interesting open problem.

Our results are obtained by a collection of reductions. Section 3 contains the reductions that constitute the proof of Theorem 1.1, and some of the reductions needed for Theorem 1.3. Section 4 contains the proof of Theorem 1.2, the remaining reductions for Theorem 1.3, and the hardness results for counting Steiner trees and connected vertex covers. A schematic representation of our reductions can be found in Figure 1.

2. PRELIMINARIES AND NOTATION

2.1. General Notation

In this paper, Δ denotes the symmetric difference and $\dot\cup$ denotes the disjoint union. For a set U and a positive integer $i \leq |U|$, we denote the family of all subsets of U of size i by $\binom{U}{i}$. In this paper, $\equiv$ will always denote congruence modulo 2, that is, $i \equiv j$ holds for integers i, j if and only if i and j have the same parity. Every assignment $\alpha\colon \{v_1, \ldots, v_n\} \to \{0, 1\}$ to n Boolean variables $v_1, \ldots, v_n$ is identified with the set $A := \{v_i \mid \alpha(v_i) = 1\} \subseteq \{v_1, \ldots, v_n\}$.

2.2. Problem definitions

Since we consider a significant number of problems in this paper, each of which has a few variants, we use the following notation for clarity. We write k-Π for problems whose input consists of set systems of sets of size at most k, or CNF formulas with clauses of width at most k. We write c-SPARSE-k-Π if, in addition, the set systems or formulas that we get as input are guaranteed to have density at most c, that is, the number of sets or clauses is at most cn, where n is the number of elements or variables.

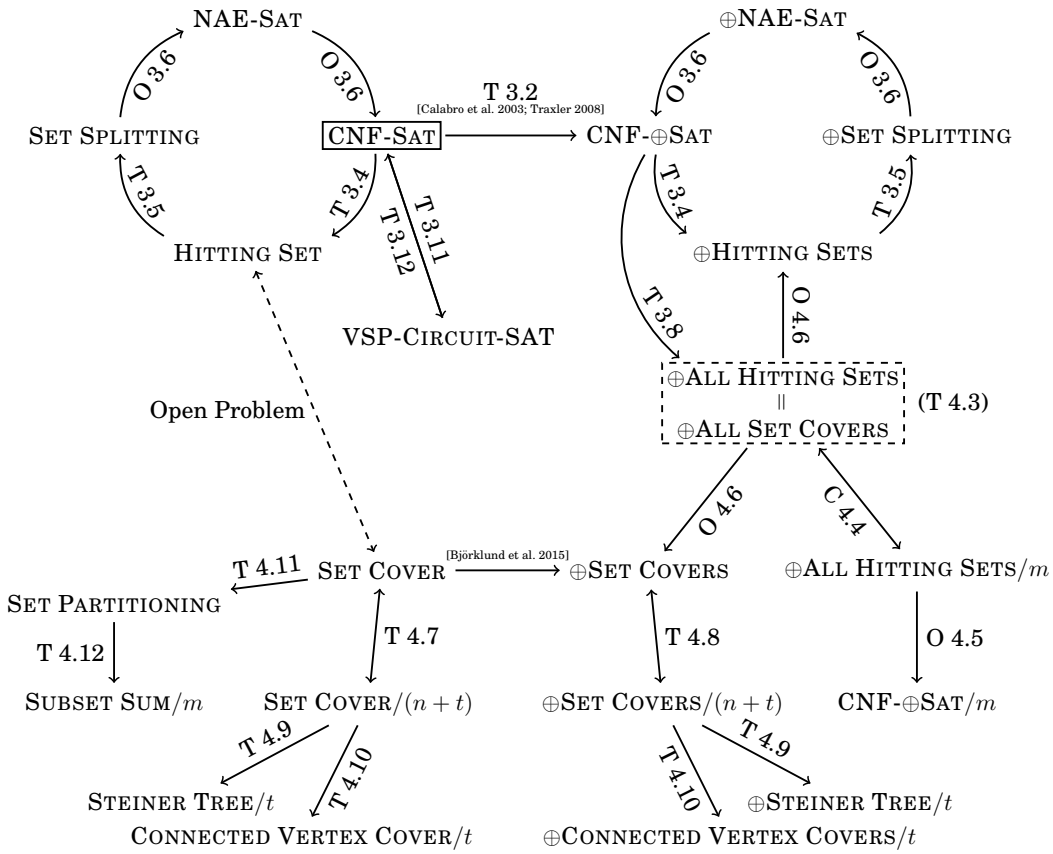

Fig. 1. Overview of the reduction graph in this paper. An arrow $\Pi/s \to \Pi'/s'$ depicts a serf-reduction from the problem Π with size-parameter s to the problem Π' with size parameter s'. Most of the problems have a secondary parameter, such as the maximum clause width or the maximum set size, which are not represented in the picture. Roughly speaking, a serf-reduction $\Pi/s \to \Pi'/s'$ implies that, if Π' can be solved in time $c^{s'} \cdot \mathrm{poly}$, then Π can be solved in time $c^{s+o(1)} \cdot \mathrm{poly}$, where the $o(1)$-term is a function whose limit is zero as the secondary parameter tends to infinity. The edge labels depict the theorem (T), corollary (C), or observation (O) that contains the formal statement of the reduction. When the size parameter s is the number of vertices or variables n, we omit it. Other parameters are: the number m of clauses, hyperedges, or the number of bits used to represent the input integers in SUBSET SUM; and the size t of the solution that we are looking for. Note that the figure suppresses details about which reductions require or preserve that the instances have bounded clause or hyperedge width, or bounded density. On the left, we have decision problems, and on the right we have parity problems; the two groups are related via the isolation lemma [Calabro et al. 2003; Traxler 2008], cf. Theorem 3.2, and via the decision-to-parity reduction of [Björklund et al. 2015]. Furthermore, we observe a cluster on the top, which contains problems for which the best-known algorithm is naïve exhaustive search; see Section 3. And there is a cluster on the bottom, which contains problems for which the best-known algorithm has a dynamic programming flavor; see Section 4. These two clusters are connected in the parity world via our "flip theorem", Theorem 4.3. In the decision world, this connection is an open problem: Does SETH imply the assumption of Theorem 1.2?

For each problem Π that we consider, we fix the canonical NP-verifier that is implicit in the way we define the problem. Then every yes-instance of Π has associated with it a set of NP-witnesses or "solutions". We write $\oplus\Pi$ for the problem of deciding whether, for a given instance, the number of its solutions is odd. For many problems, we are looking for certain subsets of size at most t, where t is given as part of the input. So when writing $\oplus\Pi$ in this case, we only count solutions of size at most t. Sometimes we want to count all solutions, not only those of at most a certain size. In this case, we add the modifier ALL to the name; for example. while $\oplus$HITTING SETS is the problem of counting modulo two all hitting sets of size at most t, the problem $\oplus$ALL HITTING SETS counts *all* hitting sets modulo two (regardless of their size).

We now state all problems that we consider in this paper, and we discuss how exactly the modifiers affect them.

2.2.1. CNF Problems. For CNF problems, the input is a CNF formula φ. We usually denote the number of variables by n and the number of clauses by m. The two basic problems that we consider are CNF-SAT and NAE-SAT.

CNF-SAT. Does φ have a satisfying assignment?
NAE-SAT. Does φ have a satisfying assignment so that (i) the first variable is set to true and (ii) each clause contains a literal set to true and a literal set to false?

We added condition (i) to NAE-SAT solely for the purpose of making its corresponding parity problem non-trivial.

Modifiers. In addition to these two basic problems, we can name new problems by adding one of the following modifiers to their names (which we do by example just for CNF-SAT).

- k-CNF-SAT is the problem in which the input formula φ is guaranteed to have at most k literals in each clause.
- c-SPARSE-k-CNF-SAT is the problem in which the input formula φ is guaranteed to have at most k literals in each clause and to have at most $m \leq c \cdot n$ clauses.

The goal of the problem remains the same in both cases, and the two modifiers only affect the promise on the input. In order to change the goal of the problem, we allow for the parity modifier, $\oplus$, to be put in front of the type of assignment that we are looking for, i.e., we have CNF-$\oplus$SAT and $\oplus$NAE-SAT. The parity modifier can be combined with one of the input modifiers.

2.2.2. Hypergraph Problems. For problems on hypergraphs, the input is a set system $\mathcal{F} \subseteq 2^U$, which consists of subsets of some universe U. The elements of U are called *vertices* and the elements of $\mathcal{F}$ are called *hyperedges*. The number of vertices is usually denoted by n and the number of hyperedges by m. The goal in all of these problems will be to find or count subsets of U that have special properties with respect to $\mathcal{F}$, or to do the dual and find or count subsets of the set system $\mathcal{F}$ that have a special property. Often there will be an additional input $t \in \mathbb{N}$ that will determine that we are looking for a subset S or a subfamily of size at most t.

We have the following four basic hypergraph problems.

HITTING SET. Does $\mathcal{F}$ have a hitting set of size at most t, that is, a subset $H \subseteq U$ with $|H| \leq t$ such that $H \cap S \neq \emptyset$ for every $S \in \mathcal{F}$?
SET COVER. Does $\mathcal{F}$ have a set cover of size at most t, that is, a subset $\mathcal{C} \subseteq \mathcal{F}$ with $|\mathcal{C}| \leq t$ such that $\bigcup_{S \in \mathcal{C}} S = U$?
SET PARTITIONING (or PERFECT SET MATCHING). Does $\mathcal{F}$ have a set partitioning of size at most t, that is, a set cover $\mathcal{C}$ such that, for every $S, S' \in \mathcal{C}$ with $S \neq S'$, we have $S \cap S' = \emptyset$?

SET SPLITTING. Is there a subset $X \subseteq U$ such that (i) the first element of the universe is a member of X and (ii), for every $S \in \mathcal{F}$, neither $S \subseteq X$ nor $S \subseteq (U-X)$?

Note that the first three problems have the additional input $t \in \mathbb{N}$, while the last problem does not. Similar to our definition of NAE-SAT, we added condition (i) in SET SPLITTING solely for the purpose of making the corresponding parity problem non-trivial.

Modifiers. The input modifiers such as in k-HITTING SET or c-SPARSE-k-HITTING SET work as before in the case of CNF problems. The number k promises that all sets S in the set system $\mathcal{F}$ will have size at most k, and the number c promises that the number m of sets is at most $c \cdot n$. We also introduce the parity modifier, $\oplus$, just as before. For example, in $\oplus$HITTING SETS, we are given t and $\mathcal{F}$, and we want to count modulo two the number of hitting sets of size at most t.

Interestingly, for parity problems, we can prove hardness results also for the case in which the input parameter t is guaranteed to be $t = n$. For decision problems, this setting of t is trivial, but the counting case turns out to be still interesting. To make this distinction clear, we add the modifier ALL in front of the object that we are counting. For clarity, we give the definition of the following modified version of HITTING SET.

$\oplus$ALL HITTING SETS
Input. A set system $\mathcal{F} \subseteq 2^U$.
Question. Does $\mathcal{F}$ have an odd number of hitting sets (of any size)?

2.2.3. Graph Problems. In graph problems, the input is a graph $G = (V, E)$ with n vertices and m edges, and often there is some additional input, such as a number $t \in \mathbb{N}$ or a set of terminals $T \subseteq V$. We consider the following basic graph problems:

CONNECTED VERTEX COVER. Does G have a connected vertex cover of size at most t, that is, a subset $X \subseteq V$ such that $|X| \leq t$, the induced subgraph $G[X]$ is connected, and $X \cap e \neq \emptyset$ holds for every edge $e \in E$?
STEINER TREE. Does G has a Steiner tree of size at most t between the terminals $T \subseteq V$, that is, is there a subset $X \subseteq V$ so that $|X| \leq t$, the induced subgraph $G[X]$ is connected, and $T \subseteq X$?

For these problems, we will only use the parity modifier. So for example, in $\oplus$CONNECTED VERTEX COVERS, we are given G and t, and we want to count modulo two the number of connected vertex covers of size at most t.

2.2.4. Other Problems. We also study the following problems.

SUBSET SUM
Input. Integers $a_1, \ldots, a_n \in \mathbb{Z}_+$ and a target integer t on m bits.
Question. Is there a subset $X \subseteq \{1, \ldots, n\}$ with $\sum_{i \in X} a_i = t$?

c-VSP-CIRCUIT-SAT
Input. A cn-size Valiant series-parallel circuit over n variables.
Question. Is there a satisfying assignment?

2.3. The optimal growth rate of a problem

Running times in this paper have the form $c^n \cdot \mathrm{poly}(m)$, where c is a nonnegative constant, m is the total size of the input, and n is a somewhat smaller parameter of the input, typically the number of variables, vertices, or elements. The constant c is the *growth rate* of the running time, and it may be different for different choices for the

parameter n. To make this parameterization explicit, we use the notation Π/n. For every such parameterized problem, we now define the number $\sigma = \sigma(\Pi/n)$.

Definition 2.1. For a parameterized problem Π/n, let $\sigma(\Pi/n)$ be the infimum over all $\sigma > 0$ such that there exists a randomized $2^{\sigma n}\mathrm{poly}(m)$-time algorithm for Π whose error probability is at most $1/3$.

The *optimal growth rate* of Π with respect to n is $C := 2^{\sigma(\Pi/n)}$. If the infimum in the definition above is a minimum, then Π has an algorithm that runs in time $C^n\mathrm{poly}(m)$ and no algorithm for Π can have a running time $c^n\mathrm{poly}(m)$ for any $c < C$. On the other hand, if the minimum does not exist, then no algorithm for Π can run in time $C^n\mathrm{poly}(m)$, but Π has a $c^n\mathrm{poly}(m)$-time algorithm for every $c > C$. We formally define the Strong Exponential Time Hypothesis (SETH) as the assertion that $\lim_{k\to\infty} \sigma(k\text{-CNF-SAT}/n) = 1$.

We remark that it is consistent with current knowledge that SETH fails and yet CNF-SAT (without restriction on the clause width) does not have $2^{\epsilon n}\mathrm{poly}(m)$-algorithms for any $\epsilon < 1$: If SETH fails, then k-CNF-SAT has, say, $k^k 1.99^n$-time algorithms for every k, which does not seem to translate to a $2^{\epsilon n}\mathrm{poly}(m)$-time algorithm for CNF-SAT for any $\epsilon < 1$.

3. ON IMPROVING BRANCHING ALGORITHMS

In this section we show that significantly faster algorithms for search problems such as HITTING SET and SET SPLITTING imply significantly faster algorithms for CNF-SAT. More precisely, we prove that the growth rates of these problems are equal, or equivalently,

$$\sigma(\text{CNF-SAT}/n) = \sigma(\text{HITTING SET}/n) = \sigma(\text{SET SPLITTING}/n).$$

We also give a reduction from CNF-$\oplus$SAT to $\oplus$ALL HITTING SETS, thus establishing a connection between the parity versions of these two problems.

3.1. Previous results for CNF-SAT

In the following few subsections, we show reductions from CNF-SAT/n to HITTING SET/n and SET SPLITTING/n. These reductions work even when the given instance of CNF-SAT/n is dense, that is, when there is no bound on the number of clauses that is linear in the number of variables. However, our starting point in Section 4 is the SPARSE-HITTING SET/n problem, where the number of sets in the set system is linear in n. For this reason we formulate our results for the sparse versions of HITTING SET/n and SET SPLITTING/n, and we develop a sparse version of SETH first.

The sparsification lemma by Impagliazzo et al. [Impagliazzo et al. 2001] is that every k-CNF formula φ can be written as the disjunction of $2^{\epsilon n}$ formulas in k-CNF, each of which has at most $c(k,\epsilon)\cdot n$ clauses. Moreover, this disjunction of sparse formulas can be computed from φ and ϵ in time $2^{\epsilon n}\cdot\mathrm{poly}(m)$. Hence, the growth rate of k-CNF-SAT for formulas of density at most $c(k,\epsilon)$ is ϵ-close to the growth rate of general k-CNF-SAT. More precisely, for every k and every $\epsilon > 0$, we have

$$\sigma\big(c\text{-SPARSE-}k\text{-CNF-SAT}/n\big) \le \sigma\big(k\text{-CNF-SAT}/n\big) \le \sigma\big(c\text{-SPARSE-}k\text{-CNF-SAT}/n\big) + \epsilon,$$

where the first inequality is trivial and the second inequality follows from the sparsification lemma. The density $c = c(k,\epsilon)$ is the *sparsification constant*, and the best known bound is $c(k,\epsilon) = (k/\epsilon)^{3k}$ [Calabro et al. 2006]. By setting $\epsilon = \epsilon(k) = o(1)$, this immediately yields the following theorem.

THEOREM 3.1 ([IMPAGLIAZZO ET AL. 2001; CALABRO ET AL. 2006]). *For every function $c = c(k) \geq (\omega(k))^{3k}$, we have*

$$\lim_{k\to\infty} \sigma\Big(k\text{-CNF-SAT}/n\Big) = \lim_{k\to\infty} \sigma\Big(c\text{-SPARSE-}k\text{-CNF-SAT}/n\Big)\,.$$

Hence, SETH is equivalent to the right-hand side being equal to 1. In [Dell et al. 2012] it was observed that the sparsification lemma can be made parsimonious, which gives the following equality for the same functions $c = c(k)$:

$$\lim_{k\to\infty} \sigma\Big(k\text{-CNF-}\oplus\text{SAT}/n\Big) = \lim_{k\to\infty} \sigma\Big(c\text{-SPARSE-}k\text{-CNF-}\oplus\text{SAT}/n\Big)\,.$$

We define $\oplus$-SETH as the assertion that these limits are equal to 1. The isolation lemmas for k-CNF formulas [Calabro et al. 2003; Traxler 2008] immediately yield that SETH implies $\oplus$-SETH. More precisely, we have the following theorem.

THEOREM 3.2 ([CALABRO ET AL. 2003; TRAXLER 2008]).

$$\lim_{k\to\infty} \sigma(k\text{-CNF-SAT}/n) \leq \lim_{k\to\infty} \sigma(k\text{-CNF-}\oplus\text{SAT}/n)\,.$$

3.2. From CNF-SAT to Hitting Set

Here we will reduce SPARSE-CNF-SAT to SPARSE-HITTING SET. For this, and also for the reduction from CNF-$\oplus$SAT to $\oplus$ALL HITTING SETS in Section 3.4, the following construction will be useful.

Given a CNF formula $\varphi = C_1 \wedge \cdots \wedge C_m$ over n variables $v_1, \ldots, v_n$ and an odd integer $p \geq 3$ that divides n, we construct the set system $\mathcal{F}_{\varphi,p} \subseteq 2^U$ as follows.

(1) Let p' be the odd integer $p' = p + 2\lceil \log_2 p \rceil$, and let $U = \{u_1, \ldots, u_{n'}\}$ with $n' = p' \cdot n/p$.
(2) Partition the variables of φ into blocks V_i of size p, i.e., $V_i := \{v_{pi+1}, \ldots, v_{p(i+1)}\}$.
(3) Partition U into blocks U_i of size p', i.e., $U_i = \{u_{p'i+1}, \ldots, u_{p'(i+1)}\}$.
(4) Choose an arbitrary injective function $\psi_i \colon 2^{V_i} \to \binom{U_i}{\lceil p'/2 \rceil}$. This exists since

$$\left|\binom{U_i}{\lceil p'/2 \rceil}\right| = \binom{p'}{\lceil p'/2 \rceil} \geq \frac{2^{p'}}{p'} \geq \frac{2^p p^2}{p + 2\lceil \log_2 p \rceil} \geq 2^p = \left|2^{V_i}\right|\,.$$

We think of ψ_i as a mapping that, given an assignment to the variables of V_i, associates with it a subset of U_i of size $\lceil p'/2 \rceil$.
(5) If $X \in \binom{U_i}{\lceil p'/2 \rceil}$ for some i, then add the set X to $\mathcal{F}_{\varphi,p}$.
(6) If $X \in \binom{U_i}{\lfloor p'/2 \rfloor}$ for some i such that $\psi_i^{-1}(\{U_i - X\}) = \emptyset$, then add the set X to $\mathcal{F}_{\varphi,p}$.
(7) For every clause C of φ, do the following:
   - Let $I = \{j \mid 1 \leq j \leq \frac{n}{p},$ and C contains a variable of block $V_j\}$;
   - For every $i \in I$, we let $\mathcal{A}_i$ be the set

$$\left\{ X_i \in \binom{U_i}{\lfloor p'/2 \rfloor} \;\middle|\; \text{some assignment in } \psi_i^{-1}(\{U_i - X_i\}) \text{ sets all literals in } C \cap V_i \text{ to false} \right\};$$

   - For every tuple $(X_i)_{i\in I}$ with $X_i \in \mathcal{A}_i$, add the set $\bigcup_{i\in I} X_i$ to $\mathcal{F}_{\varphi,p}$.

LEMMA 3.3. *For every n-variable CNF formula φ and every odd integer $p \geq 3$ that divides n, the number of satisfying assignments of φ is equal to the number of hitting sets of size $\lceil \frac{p'}{2} \rceil \frac{n}{p}$ of the set system $\mathcal{F}_{\varphi,p}$, where $p' = p + 2\lceil \log_2 p \rceil$.*

PROOF. For convenience denote $g = \frac{n}{p}$. Define $\psi \colon 2^V \to 2^U$ as $\psi(A) = \bigcup_{i=1}^{g} \psi_i(A \cap V_i)$. Note that ψ is injective, since for every i, ψ_i is injective and the V_i's partition V. Hence

to prove the lemma, it is sufficient to prove that (1) A is a satisfying assignment if and only if $\psi(A)$ is a hitting set of size $\lceil \frac{p'}{2} \rceil g$, and (2) if there is no assignment $A \subseteq V$ such that $\psi(A) = H$, than no set $H \subseteq U$ of size $\lceil \frac{p'}{2} \rceil g$ is a hitting set of $\mathcal{F}_{\varphi,p}$.

For the forward direction of (1), note that the sets added in Step 5 are hit by the pigeon-hole principle since $|\psi_i(A \cap V_i)| = \lceil \frac{p'}{2} \rceil$ and p' is odd. For the sets added in Step 6, consider the following. The set X of size $\lfloor p'/2 \rfloor$ is added because for some i, $\psi_i^{-1}(\{U_i - X\}) = \emptyset$. Thus $\psi_i(A \cap V_i)$ automatically hits X. For the sets added in Step 7, consider a clause C of φ and the associated index set I as in Step 7. Since A is a satisfying assignment of φ, there exists $i \in I$ such that A sets at least one variable in $C \cap V_i$ to true. Hence, $U_i - \psi_i(A \cap V_i) \notin \mathcal{A}_i$. On the other hand, $U_i - \psi_i(A \cap V_i)$ is the only member of $\mathcal{F}_{\varphi,p}$ that cannot be hit by $\psi(A \cap V_i)$. Therefore, all sets added in Step 7 are hit by $\psi(A)$. It is easy to check that $\psi(A)$ has size $\lceil \frac{p'}{2} \rceil g$ since there are g blocks.

For the reverse direction of (1), let A be an assignment such that $\psi(A)$ is a hitting set of size $\lceil \frac{p'}{2} \rceil g$. We show that A is a satisfying assignment of φ. Suppose for the sake of contradiction that a clause C is not satisfied by A, and let I be as defined in Step 7 for this C. Since $\psi(A)$ is a hitting set, $|\psi(A) \cap U_i| \geq \frac{p'}{2}$ for every i because it hits all sets added in Step 5. More precisely, $|\psi(A) \cap U_i| = \lceil \frac{p'}{2} \rceil$ because $|\psi(A)| = \lceil \frac{p'}{2} \rceil g$ and there are g disjoint blocks $U_1, \ldots, U_g$. Therefore, $|U_i - \psi(A)| = \lfloor \frac{p'}{2} \rfloor$, and so $U_i \cap \psi(A) = U_i - (U_i - \psi(A))$ is a member of $\mathcal{A}_i$ for every i. This means that in Step 7 the set $\bigcup_{i \in I} A_i$ with $A_i = U_i - \psi(A)$ was added, but this set is not hit by $\psi(A)$. So it contradicts that $\psi(A)$ is a hitting set.

For (2), let $H \subseteq U$ be a set of size $\lceil \frac{p'}{2} \rceil g$ and assume that there is no assignment $A \subseteq V$ such that $\psi(A) = H$. We show that H is not a hitting set of $\mathcal{F}_{\varphi,p}$. For the sake of contradiction, suppose that H is a hitting set. Then, as in the proof of the reverse direction of (1), we obtain $|H \cap U_i| = \lceil \frac{p'}{2} \rceil$ for every i. Since it hits all sets added in Step 6, we also know that $\psi_i^{-1}(\{H \cap U_i\}) \neq \emptyset$ for every i. However, this contradicts the non-existence of $A \subseteq V$ such that $\psi(A) = H$. □

THEOREM 3.4. *For every non-decreasing function $c = c(k)$, there exists a non-decreasing function $c' = c'(k')$ such that*

$$\lim_{k\to\infty} \sigma(c\text{-SPARSE-}k\text{-CNF-SAT}/n) \leq \lim_{k'\to\infty} \sigma(c'\text{-SPARSE-}k'\text{-HITTING SET}/n)\,, \textit{ and}$$
$$\lim_{k\to\infty} \sigma(c\text{-SPARSE-}k\text{-CNF-}\oplus\text{SAT}/n) \leq \lim_{k'\to\infty} \sigma(c'\text{-SPARSE-}k'\text{-}\oplus\text{HITTING SETS}/n)\,.$$

PROOF. We prove that, for any positive integer k and for any positive odd integer $p \geq 3$, there exist positive integers $k' = k'(p) := p'k$ and $c' = c'(k') := 2^{k'+1}c(k')$ such that

$$\sigma(c\text{-SPARSE-}k\text{-CNF-SAT}/n) \leq \sigma(c'\text{-SPARSE-}k'\text{-HITTING SET}/n) + O\left(\frac{\log p}{p}\right).$$

As $p \to \infty$, the right-hand side tends to the right-hand side of the inequality that we want to prove, and since the inequality holds for all k, it also holds as $k \to \infty$.

To prove the claim, we let φ be a k-CNF formula of density at most $c(k)$, and we create the set system $\mathcal{F}_{\varphi,p}$ as described above together with the desired hitting set size $t = \lceil \frac{p'}{2} \rceil \frac{n}{p}$, and we recall that $p' = p + 2\lceil \log_2 p \rceil$. For any constant p, this can clearly be done in polynomial time. By Lemma 3.3, this is a reduction from CNF-SAT to HITTING SET, and the reduction is parsimonious, that is, the number of hitting sets is exactly

equal to the number of satisfying assignments. It remains to check that the set system uses at most $c'n'$ sets, each of size at most k', and that the inequality above holds.

It is easy to see that any set in $\mathcal{F}_{\varphi,p}$ has size at most k'. Let m' be the number of sets in $\mathcal{F}_{\varphi,p}$. We observe that there are at most $2^{p'}n/p$ sets added in Step 5 and Step 6. Moreover, since each clause contains variables from at most k blocks, there are at most $2^{p'k}m$ sets added in Step 7. Therefore $m'/n' \leq m'/n \leq 2^{p'} + 2^{kp'}c(k) \leq c'(k')$ holds, where we use the monotonicity of c. This means that we can determine whether φ is satisfiable in time $2^{\sigma(c'\text{-Sparse-}k'\text{-Hitting Set}/n)n'} \cdot \mathrm{poly}(n)$, where n' is the size of the universe of $\mathcal{F}_{\varphi,p}$. Since $n' = \frac{n}{p}(p + 2\lceil \log p \rceil) = n(1 + O(\frac{\log p}{p}))$ and $\sigma \leq 1$, the claim follows. □

We remark that the proof also works when there is no restriction on the density and even when there is no restriction on the clause/set size. This is because the running time of the reduction is polynomial time for every constant p. Furthermore, the theorem trivially holds for the counting versions of the problems as well.

3.3. From Hitting Set via Set Splitting to CNF-SAT

THEOREM 3.5.

$$\lim_{k\to\infty} \sigma(k\text{-Hitting Set}/n) \leq \lim_{k\to\infty} \sigma(k\text{-Set Splitting}/n)\text{, and}$$
$$\lim_{k\to\infty} \sigma(k\text{-}\oplus\text{Hitting Sets}/n) \leq \lim_{k\to\infty} \sigma(k\text{-}\oplus\text{Set Splitting}/n)\,.$$

PROOF. It is enough to show that, for all positive integers k and p, we have

$$\sigma(k\text{-Hitting Set}/n) \leq \sigma(k'\text{-Set Splitting}/n) + \frac{\log_2(p+1)}{p}\,,$$

where $k' = \max(k+1, p+1)$. Let $(\mathcal{F}, t)$ be an instance of k-Hitting Set. We can assume that the universe U of $\mathcal{F}$ has n elements and that p divides n. Let $U = U_1 \dot\cup \ldots \dot\cup U_{n/p}$ be a partition in which each part has exactly $|U_i| = p$ elements of the universe U. Let $t_1, \ldots, t_{n/p}$ be nonnegative integers such that $\sum_{i=1}^{n/p} t_i = t$. The t_i's are our current guess for how many elements of a t-element hitting set will intersect with the U_i's. The number of ways to write t as the ordered sum of n/p nonnegative integers $t_1, \ldots, t_{n/p}$ with $0 \leq t_i \leq p$ can be bounded by $(p+1)^{n/p} = 2^{n\cdot\log(p+1)/p}$. For each choice of the t_i's, we construct an instance $\mathcal{F}'$ of k'-Set Splitting as follows.

(1) Let R (red) and B (blue) be two special elements and add the set $\{R, B\}$ to $\mathcal{F}'$.
(2) For all i with $t_i < p$ and for all $X \in \binom{U_i}{t_i+1}$, add $X \cup \{R\}$ to $\mathcal{F}'$.
(3) For every $Y \in \mathcal{F}$, add $Y \cup \{B\}$ to $\mathcal{F}'$.

Clearly $\mathcal{F}'$ can be computed in polynomial time and its universe has $n + 2$ elements. The sets added in step 2 have size at most $p + 1$ and the sets added in step 3 have size at most $k + 1$. Given an algorithm for Set Splitting, we compute $\mathcal{F}'$ for every choice of the t_i's and we decide Hitting Set in time $2^{(\epsilon+\sigma(k'\text{-Set Splitting}/n))\cdot n} \cdot \mathrm{poly}(m)$, where $\epsilon = \log(p+1)/p$. It remains to show the correctness of the reduction, i.e., that $\mathcal{F}$ has a hitting set of size at most t if and only if $\mathcal{F}'$ has a set splitting for some choice of $t_1, \ldots, t_{n/p}$.

For the completeness of the reduction, let H be a hitting set of size t and set $t_i = |U_i \cap H|$ for all i. We now observe that $H \cup \{R\}$ and its complement $(U - H) \cup \{B\}$ form a set splitting of $\mathcal{F}'$. The set $\{R, B\}$ added in step 1 is split. The sets $X \cup \{R\}$ added in step 2 are split since at least one of the $t_i + 1$ elements of $X \subseteq U_i$ is not contained in H.

Finally, the sets $Y \cup \{B\}$ added in step 3 are split since each $Y \in \mathcal{F}$ has a non-empty intersection with H.

For the soundness of the reduction, let $(S, \overline{S})$ be a set splitting of $\mathcal{F}'$ for some choice of $t_1, \ldots, t_{n/p}$. Without loss of generality, assume that R is the first vertex and thus, because of the way we defined SET SPLITTING, we will have $R \in S$. By the set added in step 1, this means that $B \in \overline{S}$. The sets added in step 2 guarantee that $U_i \cap S$ contains at most t_i elements for all i. Finally, the sets added in step 3 make sure that each set $Y \in \mathcal{F}$ has a non-empty intersection with S. Thus, $S - \{R\}$ is a hitting set of $\mathcal{F}$ and has size at most $\sum_i t_i = t$.

The claim for the parity versions follows as well since the reduction preserves the number of solutions exactly. □

OBSERVATION 3.6. *For any positive integer k we have*

$$\sigma(k\text{-SET SPLITTING}/n) \leq \sigma(k\text{-NAE-SAT}/n) \leq \sigma(k\text{-CNF-SAT}/n), \text{ and}$$
$$\sigma(k\text{-}\oplus\text{SET SPLITTING}/n) \leq \sigma(k\text{-}\oplus\text{NAE-SAT}/n) \leq \sigma(k\text{-CNF-}\oplus\text{SAT}/n)\,.$$

PROOF. For the first reduction, let $\mathcal{F}$ be an instance of k-SET SPLITTING. We construct an equivalent k-CNF formula φ as follows. For each element in the universe of $\mathcal{F}$, we add a variable, and for each set $X \in \mathcal{F}$ we add a clause in which each variable occurs positively. A characteristic function of a set splitting $U = U_1 \dot\cup U_2$ is one that assigns 1 to the elements in U_1 and 0 to the elements of U_2. Observe that the characteristic functions of set splittings of $\mathcal{F}$ stand in one-to-one correspondence to variable assignments that satisfy the NAE-SAT constraints of φ. Thus, any algorithm for k-NAE-SAT works for k-SET SPLITTING, too.

For the second reduction, let φ be a k-NAE-SAT-formula. The standard reduction to k-CNF-SAT creates two copies of every clause of φ and flips the sign of all literals in the second copies. Then any NAE-SAT-assignment of φ satisfies both copies of the clauses of φ'. On the other hand, any satisfying assignment of φ' sets a literal to true and a literal to false in each clause of φ. To make the satisfying assignments of φ' exactly the same as the NAE-assignments of φ, we furthermore add a single clause that forces the first variable of x to be set to true (recall that this requirement was part of our definition of NAE-SAT). Thus, any algorithm for k-CNF-SAT works for k-NAE-SAT, too. □

3.4. From Parity CNF-SAT to Parity All Hitting Sets

Given a CNF formula φ over n variables and clauses of size at most k and an odd integer $p \geq 3$ that divides n, we first construct the set system $\mathcal{F}_{\varphi,p} \subseteq 2^U$ as described in Section 3.2. Given the set system $\mathcal{F}_{\varphi,p} \subseteq 2^U$, we create the set system $\mathcal{F}'_{\varphi,p}$ as follows.

(8) For every block U_i:
- ◦ add a special element e_i to the universe,
- ◦ for every $X \in \binom{U_i}{\lfloor p'/2 \rfloor}$, add the set $X \cup \{e_i\}$ to the set family.

LEMMA 3.7. *The number of hitting sets of size $t = \lceil p'/2 \rceil \frac{n}{p}$ in $\mathcal{F}_{\varphi,p}$ is odd if and only if the number of* all *hitting sets in $\mathcal{F}'_{\varphi,p}$ is odd.*

PROOF. Let $g = \frac{n}{p}$. We first prove that the number of hitting sets of $\mathcal{F}_{\varphi,p}$ of size $\lceil p'/2 \rceil g$ is equal to the number of hitting sets H' of $\mathcal{F}'_{\varphi,p}$ such that $|H' \cap U_i| = \lceil \frac{p'}{2} \rceil$ for every $1 \leq i \leq g$. Suppose that H is a hitting set of $\mathcal{F}_{\varphi,p}$ of size $\lceil p'/2 \rceil g$, then it is easy to see that $H \cup \{e_1, \ldots, e_g\}$ is a hitting set of $\mathcal{F}'_{\varphi,p}$ since all the sets added in Step 8 are hit by some e_i, and indeed $|H' \cap U_i| = \lceil \frac{p'}{2} \rceil$ for every $1 \leq i \leq g$ since otherwise the set $U_i - H'$ added in Step 5 is not hit by H'. For the reverse direction, suppose H' is a

hitting set of $\mathcal{F}'_{\varphi,p}$ such that $|H' \cap U_i| = \lceil \frac{p'}{2} \rceil$ for every $1 \leq i \leq g$. Then $\{e_1, \ldots, e_g\} \subseteq H'$ since all the sets added in Step 8 are hit by H'. And hence we have a bijection between the two families of hitting sets.

For every hitting set H' of $\mathcal{F}'_{\varphi,p}$ and block U_i, we know that $|H' \cap U_i| \geq \lceil p'/2 \rceil$. So it remains to show that the number of hitting sets H' of $\mathcal{F}'_{\varphi,p}$ such that there is an $1 \leq i \leq g$ with $|H' \cap U_i| > \lceil \frac{p'}{2} \rceil$ is even. Given such a hitting set H', let $\gamma(H') = H' \Delta \{e_i\}$ where i is the smallest integer such that $|H' \cap U_i| > \lceil \frac{p'}{2} \rceil$. Obviously γ is its own inverse and $|\gamma(H') \cap U_i| > \lceil \frac{p'}{2} \rceil$ so now it remains to show that $\gamma(H')$ is also a hitting set of $\mathcal{F}'_{\varphi,p}$. To see this, notice that all sets $X \cup \{e_i\}$ added in Step 8 where $X \in \binom{U_i}{\lfloor p'/2 \rfloor}$ are hit since $|\gamma(H') \cap U_i| > \lceil \frac{p'}{2} \rceil$ and that those are the only sets containing e_i. □

THEOREM 3.8. *For every non-decreasing function $c = c(k)$, there exists a non-decreasing function $c' = c'(k')$ such that*

$$\lim_{k \to \infty} \sigma(c\text{-SPARSE-}k\text{-CNF-}\oplus\text{SAT}/n) \leq \lim_{k' \to \infty} \sigma(c'\text{-SPARSE-}k'\text{-}\oplus\text{ALL HITTING SETS}/n)\,.$$

PROOF. Let φ be an instance of c-SPARSE-k-CNF-$\oplus$SAT. First recall from the proof of Theorem 3.4 that the reduction

$$\sigma(c\text{-SPARSE-}k\text{-CNF-}\oplus\text{SAT}/n) \leq \sigma(c'\text{-SPARSE-}k'\text{-}\oplus\text{HITTING SETS}/n) + O\left(\frac{\log p}{p}\right)$$

worked by constructing the set system $\mathcal{F}_{\varphi,p}$, and that the reduction was parsimonious. Thus, when we now further move to $\mathcal{F}'_{\varphi,p}$, we have that the parity of the number of all hitting sets in $\mathcal{F}'_{\varphi,p}$ is equal to the parity of the number of hitting sets of size at most t in $\mathcal{F}_{\varphi,p}$ (by Lemma 3.7), which in turn is equal to the parity of the number of satisfying assignments to φ. Thus, this is a valid reduction from CNF-$\oplus$SAT to $\oplus$ALL HITTING SETS; since the maximum edge size k' does not increase, we just have to verify that the instance remains sparse and does not have too many more vertices.

For the density, note that, in Step 8, we add at most $2^{p'} n/p$ sets, so the density c' of $\mathcal{F}_{\varphi,p}$ goes up by at most an additive term of $2^{p'}/p$, which can be easily bounded by a function just of k'. For the running time, note that the number n' of vertices in $\mathcal{F}_{\varphi,p}$ goes up by exactly n/p', that is, the new number n'' of vertices can be bounded by $n'' \leq (1+1/p')n'$. As $p \to \infty$, this will approach $n'' \leq n'$. The claim follows because we can determine the parity of the number of hitting sets of size at most t in the set system $\mathcal{F}_{\varphi,p}$ by running the best algorithm for the corresponding problem $\oplus$ALL HITTING SETS, which runs in time $2^{\sigma(c''\text{-SPARSE-}k'\text{-}\oplus\text{ALL HITTING SETS}/n)n''} \cdot \mathrm{poly}(m)$. □

Note that conversely, an improved algorithm for CNF-$\oplus$SAT gives an improved algorithm for $\oplus$ALL HITTING SETS. This is because instances of $\oplus$ALL HITTING SETS can be viewed in a natural way a monotone CNF formulas: given a set family $\mathcal{F} \subseteq U$ we simply associate a variable with every element of U and a monotone clause for every set $S \in \mathcal{F}$.

OBSERVATION 3.9. *For all positive integers k and c, we have*

$$\sigma(c\text{-SPARSE-}k\text{-}\oplus\text{ALL HITTING SETS}/n) \leq \sigma(c\text{-SPARSE-}k\text{-CNF-}\oplus\text{SAT}/n)$$

3.5. Satisfiability for Series-Parallel Circuits

In this subsection, we show that the satisfiability of cn-size *series-parallel* circuits can be decided in time time $2^{\delta n}$ for $\delta < 1$ independent of c if and only if SETH is not true. Here the size of a circuit is the number of wires. Our proof is based on a

result of Valiant regarding paths in sparse graphs [Valiant 1977]. Calabro [Calabro 2008] discusses various notions of series-parallel graphs and provides a more complete proof of Valiant's lower bound on the size of series-parallel graphs (which he calls Valiant series-parallel graphs) that have "many" long paths. We remark that the class of Valiant series-parallel graphs is not the same as the notion of series-parallel graphs used most commonly in graph theory (see [Calabro 2008]).

In this section a *multidag* $G = (V, E)$ is a directed acyclic multigraph. Let input(G) denote the set of vertices $v \in V$ such that the indegree of v in G is zero. Similarly, let output(G) denote the set of vertices $v \in V$ such that the outdegree of v in G is zero. A *labeling* of G is a function $l\colon V \to \mathbb{N}$ such that $\forall (u, v) \in E$, $l(u) < l(v)$. A labeling l is *normal* if for all $v \in$ input(G), $l(v) = 0$ and there exists an integer $d \in \mathbb{N}$ such that for all $v \in$ output$(G) -$ input(G), $l(v) = d$. A multidag G is *Valiant series-parallel* (VSP) if it has a normal labeling l such that there exist no $(u, v), (u', v') \in E$ such that $l(u) < l(u') < l(v) < l(v')$.

We say that a boolean circuit C is a VSP *circuit* if the underlying multidag of C is a VSP graph and the indegree of every node is at most two (namely, the fan-in of each gate is at most two). Using the depth-reduction result by Valiant [Valiant 1977] and following the arguments by Calabro [Calabro 2008] and Viola [Viola 2009], we may show the following.

THEOREM 3.10. *Let C be a* VSP *circuit of size cn with n input variables. There is an algorithm A which on input C and a parameter $d \geq 1$ outputs an equivalent depth-3 unbounded fan-in OR-AND-OR circuit C' with the following properties.*

(*1*) *Fan-in of the top OR gate in C' is bounded by $2^{n/d}$.*
(*2*) *Fan-in of the bottom OR gates is bounded by $2^{2^{\mu cd}}$ where μ is an absolute constant.*
(*3*) *A runs in time $O(2^{n/d}n^{O(1)})$ if c and d are constant.*

In other words, for all $d \geq 1$, Theorem 3.10 reduces the satisfiability of a cn-size VSP circuit to that of the satisfiability of a disjunction of $2^{n/d}$ k-CNFs where $k \leq 2^{2^{\mu cd}}$ in time $O(2^{n/d}n^{O(1)})$. This implies that

$$\sigma(c\text{-VSP-CIRCUIT-SAT}/n) \leq \sigma(2^{2^{\mu cd}}\text{-CNF-SAT}/n) + \frac{1}{d}.$$

Hence, we obtain the following theorem.

THEOREM 3.11.

$$\lim_{c\to\infty} \sigma(c\text{-VSP-CIRCUIT-SAT}/n) \leq \lim_{k\to\infty} \sigma(k\text{-CNF-SAT}/n).$$

For the reverse direction, observe that a CNF formula with cn clauses, all of size at most k, can be written as a $4ck$-size VSP circuit. This observation implies that

$$\sigma(c\text{-SPARSE-}k\text{-CNF-SAT}/n) \leq \sigma(4ck\text{-VSP-CIRCUIT-SAT}/n).$$

Together with the sparsification lemma, Theorem 3.1, we obtain the following theorem.

THEOREM 3.12.

$$\lim_{k\to\infty} \sigma(k\text{-CNF-SAT}/n) \leq \lim_{c\to\infty} \sigma(c\text{-VSP-CIRCUIT-SAT}/n)\,.$$

4. ON IMPROVING DYNAMIC PROGRAMMING BASED ALGORITHMS

In this section we give some reductions that show that several dynamic programming based algorithms cannot be improved unless the growth rate of CNF-SAT can be improved. In the parity world, our starting point will be the hardness of $\oplus$ALL HITTING SETS$/n$ as proved in Theorem 3.8. More specifically, we show that

⊕ALL HITTING SETS and ⊕ALL SET COVERS are actually the *same* problem, for which we use a simple but novel property of independent sets in bipartite graphs in §4.1. In §4.2 we show that the current algorithms for ⊕STEINER TREE/t and ⊕CONNECTED VERTEX COVERS/t are at least as hard to improve as the algorithm for ⊕ALL SET COVERS/n. Motivated by these facts, we concoct the hypothesis that the growth rate 2 of the best known algorithm for SET COVER cannot be improved, and we show similar implications for the problems STEINER TREE/t and CONNECTED VERTEX COVER/k, SET PARTITIONING and SUBSET SUM.

4.1. The flip: Parity Hitting Set equals Parity Set Cover

It is well known that the Hitting Set and the Set Cover problem are dual to each other: The hitting sets of any set family $\mathcal{F}$ are in one-to-one correspondence with the set covers of its dual set family $\mathcal{F}^*$. Here the dual is defined by flipping the roles of sets and elements: in $\mathcal{F}^*$, every element becomes a set and every set becomes an element, but we preserve all incidences between them.

OBSERVATION 4.1. *For all set families $\mathcal{F}$, we have*

$$\oplus\text{ALL HITTING SETS}(\mathcal{F}) = \oplus\text{ALL SET COVERS}(\mathcal{F}^*)\,.$$

We demonstrate now that, in the parity world, the duality between hitting set and set cover is very strong: Indeed, the two parities are equal even without going to the dual set system! For this, we first state the following intermediate step.

LEMMA 4.2. *Let $G = (A \cup B, E)$ be a bipartite graph, then the number of independent sets of G modulo two is equal to $|\{X \subseteq A : N(X) = B\}| \bmod 2$.*

PROOF. Grouping on their intersection with A, the number of independent sets of G is equal to

$$\sum_{X\subseteq A} 2^{|B-N(X)|} \equiv \sum_{\substack{X\subseteq A\\ |B-N(X)|=0}} 2^0 \equiv |\{X \subseteq A : N(X) = B\}| \bmod 2\,.$$

Thus, the lemma holds. □

This lemma was inspired by a non-modular variant from [Nederlof and van Rooij 2010, Lemma 2] (see also [van Rooij 2011, Proposition 9.1]). We now show that, for any set system, the parity of the number of hitting sets is always equal to the parity of the number of set covers.

THEOREM 4.3 (FLIP THEOREM). ⊕ALL HITTING SETS = ⊕ALL SET COVERS.

PROOF. Let $\mathcal{F} \subseteq 2^U$ be a set system, let $G = (\mathcal{F}, U, E)$ be the bipartite graph where $(S, e) \in E$ if and only if $e \in S$. Note that the number of hitting sets of $\mathcal{F}$ is equal to $|\{X \subseteq U : N(X) = \mathcal{F}\}|$. Then by Lemma 4.2, the number of hitting sets is equal to the number of independent sets of G modulo 2. And similarly, since the lemma is symmetric with respect to the two color classes of the bipartite graph, the number of set covers of $\mathcal{F}$ is also equal to the number of independent sets of G modulo 2. Thus all three parities are equal. □

Let us emphasize once again that the problem ⊕ALL HITTING SETS is equal to the problem ⊕ALL SET COVERS. If, in the following, we use two different names, we do so only because the view of one or the other is more convenient for us.

The duality observation and the theorem above give rise to the following curious corollary.

COROLLARY 4.4. $\sigma(\oplus\text{ALL HITTING SETS}/n) = \sigma(\oplus\text{ALL HITTING SETS}/m)$

That is, $\oplus$ALL HITTING SETS has a $1.99^n \cdot \mathrm{poly}(m+n)$ algorithm if and only it has a $1.99^m \cdot \mathrm{poly}(m+n)$ algorithm. Since hitting sets can be seen as satisfying assignments of a monotone CNF formula, we can also formulate an analogue of Observation 3.9.

OBSERVATION 4.5. $\sigma(\oplus\text{ALL HITTING SETS}/m) \leq \sigma(\text{CNF-}\oplus\text{SAT}/m)$.

Putting all things together, we proved that a $1.99^m \cdot \mathrm{poly}(m+n)$ algorithm for CNF-$\oplus$SAT implies a $1.99^n \cdot \mathrm{poly}(m+n)$ time algorithm for the same problem, and thus such an algorithm would violate SETH.

We finish this discussion with one more observation: We can always reduce from the problem $\oplus$ALL HITTING SETS to $\oplus$HITTING SETS and to $\oplus$SET COVERS.

OBSERVATION 4.6. *For all size parameters s of $\oplus$ALL HITTING SETS, we have*

$$\sigma(\oplus\text{ALL HITTING SETS}/s) \leq \sigma(\oplus\text{HITTING SETS}/s), \textit{ and}$$
$$\sigma(\oplus\text{ALL HITTING SETS}/s) \leq \sigma(\oplus\text{SET COVERS}/s)\,.$$

PROOF. Note that $\oplus$ALL HITTING SETS is equal to the problem $\oplus$HITTING SETS in which the size t of the hitting sets we are counting is fixed to $t = n$, i.e., we count all hitting sets. Then any algorithm for $\oplus$HITTING SETS will immediately work for $\oplus$ALL HITTING SETS as well. The analogous argument applies to $\oplus$SET COVERS. □

4.2. From Set Cover to Steiner Tree and Connected Vertex Cover

In this subsection we will give reductions from SET COVER/n to STEINER TREE/t and CONNECTED VERTEX COVER/k. We transfer the reductions to the parity versions SET COVER/n, $\oplus$STEINER TREE/t, and $\oplus$CONNECTED VERTEX COVERS/k. For the reduction, we first need an intermediate result, showing that SET COVER/$(n+t)$, that is, SET COVER parameterized by the sum of the size of the universe and solution size, is as hard as SET COVER/n (and similarly for $\oplus$SET COVERS/n and $\oplus$SET COVERS/$(n+t)$). Once we have this intermediate result, the reductions to the $\oplus$STEINER TREE/t and $\oplus$CONNECTED VERTEX COVERS/k problems follow more easily.

THEOREM 4.7.

$$\lim_{k\to\infty} \sigma(k\text{-SET COVER}/n) = \lim_{k\to\infty} \sigma(k\text{-SET COVER}/(n+t))\,.$$

PROOF. The case $\geq$ follows from the basic fact that increasing the size parameter cannot increase the running time relative to the parameter.

To prove $\leq$, we use the "powering" technique for SET COVER: for each constant $\alpha > 0$, we transform an instance $(\mathcal{F}, U, t)$ of k-SET COVER into an instance of k'-SET COVER, for some positive integer k', where the size t' of the solution in the resulting p'-SET COVER instances is at most $\alpha|U|$, without changing the universe size.

Without loss of generality, we assume that $t \leq |U|$. Consider any $\alpha > 0$. Let q be the smallest positive integer such that $\frac{1}{q} \leq \alpha$. We may assume that t is divisible by q, since otherwise we may add at most q additional elements to the universe U and singleton sets to the family $\mathcal{F}$. We form a family $\mathcal{F}'$ of all unions of exactly q sets from $\mathcal{F}$, that is for each of $\binom{|\mathcal{F}|}{q}$ choices of q sets $S_1, \ldots, S_q \in \mathcal{F}$ we add to $\mathcal{F}'$ the set $\bigcup_{i=1}^{q} S_i$. Note that since q is a constant we can create $\mathcal{F}'$ in polynomial time. We set $t' = t/q \leq |U|/q \leq \alpha|U|$. It is easy to see that $(\mathcal{F}, U, t)$ is a YES-instance of k-SET COVER if and only if $(\mathcal{F}', U, t')$ is a YES-instance of qk-SET COVER. □

Observe that in the proof above, because of the grouping of q sets, one solution for the initial instance may correspond to several solutions in the resulting instance. For this reason the counting variant of the above reduction is much more technically involved.

THEOREM 4.8. *For every function $c = c(k)$, we have*

$$\lim_{k\to\infty} \sigma(c\text{-SPARSE-}k\text{-}\oplus\text{SET COVERS}/n) \leq \lim_{k'\to\infty} \sigma(k'\text{-}\oplus\text{SET COVERS}/(n+t))\,.$$

The reverse $\sigma(c\text{-SPARSE-}k\text{-}\oplus\text{SET COVERS}/n) \geq \sigma(c\text{-SPARSE-}k\text{-}\oplus\text{SET COVERS}/(n+t))$ holds trivially for all k and c. The proof of Theorem 4.8 is quite involved, and we postpone it to the end of this section. Instead, we will first look at some of its consequences.

THEOREM 4.9.

$$\lim_{k\to\infty} \sigma(k\text{-SET COVER}/(n+t)) \leq \sigma(\text{STEINER TREE}/t)\,, \textit{ and}$$
$$\lim_{k\to\infty} \sigma(k\text{-}\oplus\text{SET COVERS}/(n+t)) \leq \sigma(\oplus\text{STEINER TREE}/t)\,.$$

PROOF. Given an instance of SET COVER consisting of a set system $(\mathcal{F}, U)$ and integer i, let G' be the graph obtained from the incidence graph of $(\mathcal{F}, U)$ by adding a vertex s universal to $\mathcal{F}$ with a pendant vertex u, and define the terminal set to be $U \cup \{u\}$. It is easy to see that the number of Steiner trees with $|U| + i + 1$ edges is equal to the number of set covers of $(\mathcal{F}, U)$ of size i. Hence the theorem follows. □

THEOREM 4.10.

$$\lim_{k\to\infty} \sigma(k\text{-SET COVER}/(n+t)) \leq \sigma(\text{CONNECTED VERTEX COVER}/t)\,, \textit{ and}$$
$$\lim_{k\to\infty} \sigma(k\text{-}\oplus\text{SET COVERS}/(n+t)) \leq \sigma(\oplus\text{CONNECTED VERTEX COVERS}/t)\,.$$

PROOF. Given an instance $(\mathcal{F}, U, t)$ of SET COVER, we create an instance of CONNECTED VERTEX COVER with G being obtained from the incidence graph of $(\mathcal{F}, U)$ by adding a vertex s adjacent to all vertices corresponding to sets and adding pendant vertices for every element of $U \cup \{s\}$. Moreover let $t' = t + |U| + 1$ in the CONNECTED VERTEX COVER instance.

It is easy to see that for every i, there exists a set cover of $(\mathcal{F}, U)$ of size $i \leq t$ if and only if there exists a connected vertex cover of G of size at most $i + |U| + 1 \leq t'$ since we can take without loss of optimality all vertices having a pendant vertex, and then connecting these vertices is equivalent to covering all elements of U with sets in $\mathcal{F}$. Hence, by using an algorithm for CONNECTED VERTEX COVER, we obtain an $O(2^{\sigma(\text{CONNECTED VERTEX COVER}/t)t'} n^{O(1)}) = O(2^{\sigma(\text{CONNECTED VERTEX COVER}/t)(|U|+t)} n^{O(1)})$ time algorithm for p-SET COVER.

For the parity case, let us study the number of connected vertex covers of size j of G for every j. Similarly to the previous case, note that for any connected vertex cover C, $C \cap \mathcal{F}$ must be a set cover of $(\mathcal{F}, U)$ by the connectivity requirement. Hence we group all connected vertex covers in G depending on which set cover in $(\mathcal{F}, U)$ their intersection with $\mathcal{F}$ is. Let c_j be the number of connected vertex covers of G of size j and s_i be the number of set covers of size i in $(\mathcal{F}, U)$, then:

$$c_j = \sum_{i=1}^{j-|U|-1} s_i \binom{|U|+1}{j-i-|U|-1}\,.$$

Now the number s_i modulo 2 can be determined in polynomial time once $(c_1, \ldots, c_{i+|U|+1})$ modulo 2 are computed by recovering s_1 up to s_i in increasing order, since for $i = j - |U| - 1$ we have $\binom{|U|+1}{j-i-|U|-1} = 1$.

Thus, if we can compute the parity of the number of connected vertex covers of size n in $O(2^{\sigma(\text{CONNECTED VERTEX COVER}/t)t'} n^{O(1)})$ time, we can compute the parity of all

$(c_1, \ldots, c_{i+|U|+1})$ in $O(2^{\sigma(\text{CONNECTED VERTEX COVER}/t)(|U|+t)} n^{O(1)})$ time, and hence the parity of s_i. □

4.3. From Set Cover via Set Partitioning to Subset Sum

THEOREM 4.11.

$$\lim_{p\to\infty} \sigma(p\text{-SET COVER}/n) \leq \lim_{p\to\infty} \sigma(p\text{-SET PARTITIONING}/n).$$

PROOF. Let $(\mathcal{F}, t)$ be an instance of p-SET COVER. Create an instance $(\mathcal{F}', t)$ of p-SET PARTITIONING by for every $S \in \mathcal{F}$ adding all subsets of S to $\mathcal{F}'$. Clearly $(\mathcal{F}', t)$ has a set partitioning of size at most t if and only if $(\mathcal{F}, t)$ has a set cover of size at most t. Since the size of the sets in $\mathcal{F}$ is bounded by p, the reduction runs in polynomial time. □

THEOREM 4.12.

$$\lim_{k\to\infty} \sigma(k\text{-SET PARTITIONING}/n) \leq \sigma(\text{SUBSET SUM}/m).$$

PROOF. Let $\mathcal{F} \subseteq 2^U$ be an instance of k-SET PARTITIONING. We iterate over all potential sizes $1 \leq t_0 \leq n$ of the solution for the SET PARTITIONING problem.

We create an instance of SUBSET SUM as follows. Let the target integer t for SUBSET SUM have a bit expansion consisting of three fields. First, as the most significant bits, a field coding the value of t_0, to check the cardinality of the solution $\mathcal{C} \subseteq \mathcal{F}$; second, a field of length $\log_2 t_0 + \log_2 n$ containing the value n, to check the total size of all sets in $\mathcal{C}$; finally, a field of length $\log_2 t_0 + n$ containing n ones. The paddings of length $\log_2 t_0$ serve to isolate the fields from each other. For every $S_i \in \mathcal{F}$, we create an integer a_i with the same field division as t, where the first field encodes 1, the second field encodes $|S_i|$, and the third field contains a one in position j if and only if $u_j \in S_i$. We argue that the resulting SUBSET SUM instance is a YES-instance if and only if $\mathcal{F}$ contains a partitioning of U using exactly t_0 sets.

Clearly, if $\mathcal{C} \subseteq \mathcal{F}$ partitions U and $|\mathcal{C}| = t_0$, then the integers a_i corresponding to $S_i \in \mathcal{C}$ sum to t. The first field sums to t_0 by cardinality of $\mathcal{C}$, the second sums to n, and in the third field the non-zero digits are simply partitioned among the a_i.

So let A be a collection of integers a_i that sum to t. By the first field, we have $|A| \leq t_0$; thus the padding of length $\log t_0$ is enough to isolate the fields, and we have $|A| = t_0$. By the same argument on the second field, the sum over all $a_i \in A$ of the number of non-zero bits in the third field is exactly n. Under these conditions, the only way that the third field can actually contain n true bits is if the true bits in the third field are partitioned among the a_i. Indeed, since the total number of non-zero bits in the third field among the numbers a_i is n, restricted to the third field we can rewrite the sum as $\sum_{i\in[n]} 2^{e_i} = 2^n - 1$, where $e_i \in \{0, \ldots, n-1\}$ for each $i \in [n]$. But $2^n - 1$ has a unique description as the sum of n powers of 2, which requires all values e_i to be distinct. Hence the non-zero bits in the third field are partitioned among the a_i, and $\mathcal{C} = \{S_i \mid a_i \in A\}$ is a set partitioning of U of cardinality exactly t_0.

By looping over all $1 \leq t_0 \leq t$ for the SET PARTITIONING instance, this solves the problem. Note that the length of the bit string t is $n + O(\log n)$, which disappears into the asymptotics. □

4.4. Proof of Theorem 4.8

As a proof we present a reduction which for fixed $\alpha > 0$ transforms an instance $(\mathcal{F}', U')$ of c-SPARSE-k-$\oplus$ALL SET COVERS into polynomially many instances of the k'-$\oplus$SET COVERS problem, for some positive integer k', where the size t of the solution in the resulting k'-$\oplus$SET COVERS instances is at most $\alpha|U'|$.

In order to find the parity of the number of all set covers of the instance $(\mathcal{F}', U')$ we find the parity of the number of set covers of a particular size. That is we iterate over all possible sizes $j = 1, \ldots, |\mathcal{F}'|$ of a set cover. Let us assume that we want to find the parity of the number of set covers of size j and for each positive integer $j' < j$ we know the parity of the number of set covers of $(\mathcal{F}', U')$ of size j'. Let q be the smallest power of two satisfying $\frac{|\mathcal{F}'|}{q} + 2 \leq \alpha|U'|$. We assume that $\alpha|U'| \geq 3$ since otherwise the instance is small and we can solve it by brute force (recall that α is a given constant). Observe that q is upper bounded by a constant independent of $|U'|$ since $|\mathcal{F}'| \leq c|U'|$.

We create a temporary set system $(\mathcal{F}_0, U_0)$ to ensure that the size of the set covers we are looking for is divisible by q. Let $r = j \bmod q$. We make $(\mathcal{F}_0, U_0)$ by taking the set system $(\mathcal{F}', U')$ and adding $q - r$ new elements to the universe U_0 and also $q - r$ singleton sets of the new elements to the family $\mathcal{F}_0$. Now we are looking for the parity of the number of set covers of size $j_0 = j + (q - r)$ in $(\mathcal{F}_0, U_0)$. Observe that for each $j' < j_0$ we know the parity of the number of set covers of size j' in $(\mathcal{F}_0, U_0)$ since it is equal to the parity of set covers of $(\mathcal{F}', U')$ of size $j' - (q - r) < j$ which we already know.

To obtain a k'-$\oplus$ALL SET COVERS instance we set $U^* = U_0$ and we form a family $\mathcal{F}^*$ of all unions of exactly q sets from $\mathcal{F}_0$, that is for each of $\binom{|\mathcal{F}_0|}{q}$ choices of q sets $S_1, \ldots, S_q \in \mathcal{F}_0$ we add to $\mathcal{F}^*$ the set $\bigcup_{i=1}^{q} S_i$ (note that $\mathcal{F}^*$ might be a multiset). Finally we set $t^* = j_0/q$ which is an integer since $j + (q - r)$ is divisible by q. Observe that $t^* \leq \frac{j}{q} + 1 \leq \alpha|U'| - 1$, by the definition of q, but $(\mathcal{F}^*, U^*, t^*)$ might not be a proper instance of kq-$\oplus$ALL SET COVERS, since $\mathcal{F}^*$ could be a multiset. Note that each subset of U^* appears in $\mathcal{F}^*$ at most $(2^{kq})^q = 2^{kq^2}$ times, since $\mathcal{F}_0$ has no duplicates and each set in $\mathcal{F}^*$ is a union of exactly q sets from $\mathcal{F}_0$. To overcome this technical obstacle we make a new instance $(\mathcal{F}, U, t)$, where as U we take U^* with $z = 1 + kq^2$ elements added, $U = U^* \cup \{e_1, \ldots, e_z\}$. We use elements $\{e_1, \ldots, e_{z-1}\}$ to make sets from $\mathcal{F}^*$ different in $\mathcal{F}$ by taking a different subset of $\{e_1, \ldots, e_{z-1}\}$ for duplicates. Additionally we add one set $\{e_1, \ldots, e_z\}$ to the family $\mathcal{F}$ and set $t = t^* + 1$. In this way we obtain $(\mathcal{F}, U, t)$, that is a proper $(kq + z)$-$\oplus$ALL SET COVERS instance and $t = t^* + 1 \leq \alpha|U'|$. Observe that in the final instance we have $|U| \leq n + q + z$ and $|\mathcal{F}| \leq (cn + q)^q + 1$, which is a polynomial since k, c, q and z are constants.

To summarize the reduction, we have taken an instance of c-SPARSE-k-$\oplus$ALL SET COVERS and iterated over the size of solution. Next we made the size divisible by q by adding additional elements to the universe and created a multiset family $\mathcal{F}^*$ from which we made a set family by differentiating identical sets with additional elements of the universe. Our goal was to decide whether the k-$\oplus$ALL SET COVERS instance $(\mathcal{F}', U')$ (for $k' = kq + z$) has an odd number of set covers, which means that we want to control the correspondence between the parity of the number of solutions in each part of the construction. Observe that the only step of the construction which has nontrivial correspondence between the number of solutions of the former and the latter instance is the grouping step where we transform an instance $(\mathcal{F}_0, U_0, j_0)$ into a multiset instance $(\mathcal{F}^*, U^*, t^*)$.

Hence we assume that we know the parity of the number of set covers of size $t^* = j_0/q$ in $(\mathcal{F}^*, U^*)$ as well as the parity of the number of set covers of size j' for each $j' < j_0$ in $(\mathcal{F}_0, U_0)$. Our objective is to compute the parity of the number of set covers of size j_0 in $(\mathcal{F}_0, U_0)$ in polynomial time and for this reason we introduce a few definitions and lemmas. Recall that each set in $\mathcal{F}^*$ corresponds to a union of exactly q sets in $\mathcal{F}_0$ and let $\Gamma \colon \mathcal{F}^* \to 2^{\mathcal{F}_0}$ be a function that for each set in $\mathcal{F}^*$ assigns a family of exactly q sets from $\mathcal{F}_0$ that it was made of. Moreover let $\mathcal{S}^* \subseteq 2^{\mathcal{F}^*}$ be the family of set covers of size t^* in $(\mathcal{F}^*, U^*)$ and let $\mathcal{S}_0 \subseteq 2^{\mathcal{F}_0}$ be the set of set covers of size *at most* j_0 in $(\mathcal{F}_0, U_0)$.

We construct a mapping $\Phi\colon \mathcal{S}^* \to \mathcal{S}_0$ which maps each set cover $A \in \mathcal{S}^*$ to a set cover $A_0 \in \mathcal{S}_0$ such that A_0 is exactly the set of sets from $\mathcal{F}_0$ used in the t^* unions of q sets from $\mathcal{F}_0$, that is $\Phi(A) = \bigcup_{X\in A} \Gamma(X)$. In the following lemma we prove that for a set cover $A_0 \in \mathcal{S}_0$ the size of $\Phi^{-1}(A_0)$ depends solely on the size of A_0.

LEMMA 4.13. *Let $A_0, B_0 \in \mathcal{S}_0$ such that $|A_0| = |B_0|$. Then $|\Phi^{-1}(A_0)| = |\Phi^{-1}(B_0)|$.*

PROOF. Let $A_0 = \{X_1, \ldots, X_a\}$ be a set from $\mathcal{S}_0$, where each $X_i \in \mathcal{F}_0$. Observe that for any $A \in \mathcal{S}^*$ we have $\Phi(A) = A_0$ if and only if $\bigcup_{i=1}^{a} \Gamma(X_i) = A$. Consequently $|\Phi^{-1}(A_0)|$ is equal to the number of set covers of size t^* in the set system $(\binom{A_0}{q}, A_0)$ and hence $|\Phi^{-1}(A_0)|$ depends only on the size of A_0. □

Now we prove that for each set cover $A_0 \in \mathcal{S}_0$ of size j_0 an odd number of set covers from $\mathcal{S}^*$ is mapped by Φ to A_0.

LEMMA 4.14. *For any pair of nonnegative integers a, b such that $b \leq a$ the binomial coefficient $\binom{a}{b}$ is odd if and only if $\operatorname{ones}(b) \subseteq \operatorname{ones}(a)$, where $\operatorname{ones}(x)$ is the set of indices containing ones in the binary representation of x.*

PROOF. For a nonnegative integer x, we denote by $f(x)$ the greatest integer i such that $x!$ is divisible by 2^i. It is easy to see that

$$f(x) = \sum_{i\geq 1} \left\lfloor \frac{x}{2^i} \right\rfloor$$

since for each i there are $\lfloor \frac{x}{2^i} \rfloor$ values from 1 to x that are divisible by 2^i. Then, we observe

$$f(x) = \sum_{i\geq 1} \left\lfloor \frac{x}{2^i} \right\rfloor = \left(\sum_{i\geq 1} \frac{x}{2^i}\right) - \frac{1}{2} \cdot \left|\left\{i \geq 1 : \left\lfloor \frac{x}{2^{i-1}} \right\rfloor \text{ is odd}\right\}\right| = \left(\sum_{i\geq 1} \frac{x}{2^i}\right) - \frac{|\operatorname{ones}(x)|}{2}.$$

Since $\binom{a}{b} = \frac{a!}{b!(a-b)!}$ we infer that $\binom{a}{b}$ is odd if and only if $f(a) = f(b) + f(a-b)$, which by the above formula is equivalent to $|\operatorname{ones}(a)| = |\operatorname{ones}(b)| + |\operatorname{ones}(a-b)|$. However for any nonnegative integers x, y we have $|\operatorname{ones}(x+y)| \leq |\operatorname{ones}(x)| + |\operatorname{ones}(y)|$ and moreover $|\operatorname{ones}(x+y)| = |\operatorname{ones}(x)| + |\operatorname{ones}(y)|$ if and only if there are no carry-operations when adding x to y, which is equivalent to $\operatorname{ones}(x) \cap \operatorname{ones}(y) = \emptyset$.

Therefore by setting $x = b$ and $y = a-b$ we infer that $\binom{a}{b}$ is odd if and only if $\operatorname{ones}(b) \cap \operatorname{ones}(a-b) = \emptyset$ which is equivalent to $\operatorname{ones}(b) \subseteq \operatorname{ones}(a)$ and the lemma follows. □

LEMMA 4.15. *Let $A_0 \in \mathcal{S}_0$ such that $|A_0| = j_0$ then $|\Phi^{-1}(A_0)|$ is odd.*

PROOF. Since $|\Phi^{-1}(A_0)|$ is equal to the number of set covers of size t^* in the set system $(\binom{A_0}{q}, A_0)$ and $|A_0| = j_0 = t^*q$ we infer that $|\Phi^{-1}(A_0)|$ is equal to the number of unordered partitions of A_0 into sets of size q. Hence $|\Phi^{-1}(A_0)| = \prod_{i=0}^{t^*-1} \binom{j_0-1-iq}{q-1}$. Since j_0 is divisible by q and q is a power of two using Lemma 4.14 we have $|\Phi^{-1}(A_0)| \equiv 1 \pmod 2$. □

For $j = 1, \ldots, j_0$ by s_j let us denote the parity of the number of set covers of $(\mathcal{F}_0, U_0)$ of size j. Recall that we know the value of s_j for each $j < j_0$ and we want to compute s_{j_0} knowing also $|\mathcal{S}^*| \bmod 2$. By Lemma 4.13 we can define d_j for $j = 1, \ldots, j_0$, that is the value of $|\Phi^{-1}(A_0)| \bmod 2$ for a set $A_0 \in \mathcal{S}_0$ of size j. By Lemma 4.15 we know that

d_{j_0} equals one. Thus we have the following congruence modulo 2.

$$|\mathcal{S}^*| \;=\; \sum_{A_0 \in \mathcal{S}_0} |\Phi^{-1}(A_0)| \equiv \sum_{j=1}^{j_0} s_j d_j = s_{j_0} + \sum_{j=1}^{j_0-1} s_j d_j \,.$$

Hence knowing $|\mathcal{S}^*| \bmod 2$ and all values s_j for $j < j_0$ in order to compute s_{j_0} it is enough to compute all the values d_j, what we can do in polynomial time thanks to the following lemma.

LEMMA 4.16. *For each $j = 1, \ldots, j_0$ we can calculate the value of d_j in polynomial time.*

PROOF. Again we use that fact that for a set $A_0 \in \mathcal{S}_0$ we have that $|\Phi^{-1}(A_0)|$ is equal to the number set covers of size t^* in the set system $(\binom{A_0}{q}, A_0)$. Using the inclusion-exclusion principle modulo two we obtain the following formula when $|A_0| = j$.

$$|\Phi^{-1}(A_0)| \equiv \sum_{X \subseteq A_0} \left| \left\{ \mathcal{H} \subseteq \binom{X}{q} \middle| |\mathcal{H}| = t^* \right\} \right| = \sum_{i=0}^{j} \binom{j}{i} \binom{\binom{i}{q}}{t^*},$$

Where the second equality follows by grouping all summands $X \subseteq A_0$ with $|X| = i$ for every $0 \leq i \leq |A_0|$. □

Consequently, by solving a polynomial of n number of instances of the k'-$\oplus$SET COVERS problem with universe size bounded by $n + q + z$ and set family size bounded by $(cn + q)^q + 1$, we verify whether the initial set system $\mathcal{F}' \subseteq 2^{U'}$ has an odd number of set covers, which finishes the proof of Theorem 4.8. □

5. SUMMARY AND OPEN PROBLEMS

We have shown that the exponential time complexity of a number of basic problems is strongly interconnected. Specifically, our results imply that the optimal growth rates of a number of problems are in fact asymptotically equal. Assuming SETH, our results imply tight lower bounds on the growth rates for a number of search problems whose growth rates are achieved by naïve brute force algorithms. For problems solvable by dynamic programming, we gave tight lower bounds assuming that the optimal growth rate of SET COVER is achieved by its known dynamic programming algorithm. Finally, we connected the two types of results by showing that SETH implies tight lower bounds on the optimal growth rates of corresponding parity variants. We conclude our work with some open problems.

(1) Is it possible to rule out an algorithm for SET COVER with running time $2^{\epsilon n} m^{O(1)}$, $\epsilon < 1$, assuming SETH?
(2) Is it possible to rule out an algorithm for GRAPH COLORING with running time $2^{\epsilon n}$, $\epsilon < 1$, assuming SETH? What about a lower bound for GRAPH COLORING under the assumption that there does not exist a $\delta < 1$ such that SET COVER with sets of size at most k has a $O(2^{\delta n} m^{O(1)})$ time algorithm for every k?
(3) Is it possible to rule out an algorithm that *counts* the number of proper c-colorings of an input graph in time $2^{\epsilon n}$, $\epsilon < 1$ assuming $\oplus$-SETH?
(4) Assuming SETH, is it possible to rule out an algorithm with running time $2^{\epsilon n} n^{O(1)}$, $\epsilon < 1$ for the satisfiability of circuits with at most cn gates of *unbounded fan in*, for a concrete constant c?
(5) Assuming SETH, is it possible to rule out an algorithm with running time $O(c^n)$ for 3-CNF-SAT for a concrete constant c?

REFERENCES

Richard Bellman. 1962. Dynamic programming treatment of the travelling salesman problem. *J. ACM* 9, 1 (1962), 61–63. DOI:http://dx.doi.org/10.1145/321105.321111

Andreas Björklund, Holger Dell, and Thore Husfeldt. 2015. The parity of set systems under random restrictions with applications to exponential time problems. In *Proceedings of the 42nd Internationcal Colloquium on Automata, Languages and Programming, ICALP 2015*. 231–242. DOI:http://dx.doi.org/10.1007/978-3-662-47672-7_19

Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. 2007. Fourier meets Möbius: Fast subset convolution. In *Proceedings of the 39th ACM Symposium on Theory of Computing, STOC 2007*. 67–74. DOI:http://dx.doi.org/10.1145/1250790.1250801

Andreas Björklund, Thore Husfeldt, and Mikko Koivisto. 2009. Set partitioning via inclusion-exclusion. *SIAM J. Comput.* 39, 2 (2009), 546–563. DOI:http://dx.doi.org/10.1137/070683933

Chris Calabro. 2008. *A lower bound on the size of series-parallel graphs dense in long paths*. Tech report TR08-110. Electronic Colloquium on Computational Complexity (ECCC). http://eccc.hpi-web.de/eccc-reports/2008/TR08-110/

Chris Calabro, Russell Impagliazzo, Valentine Kabanets, and Ramamohan Paturi. 2003. The complexity of unique k-SAT: An isolation lemma for k-CNFs. In *Proceedings of the 18th Annual IEEE Conference on Computational Complexity, CCC 2003*. 135. DOI:http://dx.doi.org/10.1109/CCC.2003.1214416

Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. 2006. A duality between clause width and clause density for SAT. In *Proceedings of the 21th Annual IEEE Conference on Computational Complexity, CCC 2006*. 252–260. DOI:http://dx.doi.org/10.1109/CCC.2006.6

Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. 2009. The complexity of satisfiability of small depth circuits. In *Proceedings of the 4th International Workshop on Parameterized and Exact Computation, IWPEC 2009*. 75–85. DOI:http://dx.doi.org/10.1007/978-3-642-11269-0_6

Jianer Chen, Benny Chor, Mike Fellows, Xiuzhen Huang, David W. Juedes, Iyad A. Kanj, and Ge Xia. 2005. Tight lower bounds for certain parameterized NP-hard problems. *Information and Computing* 201, 2 (2005), 216–231. DOI:http://dx.doi.org/10.1016/j.ic.2005.05.001

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. *Introduction to algorithms* (third ed.). MIT Press.

Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. 2011. Solving connectivity problems parameterized by treewidth in single exponential time. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science, FOCS 2011*. 150–159. DOI:http://dx.doi.org/10.1109/FOCS.2011.23

Holger Dell, Thore Husfeldt, Dániel Marx, Nina Taslaman, and Martin Wahlén. 2012+. Exponential time complexity of the permanent and the Tutte polynomial. *ACM Transactions on Algorithms* (2012+). To appear.

Fedor V. Fomin, Fabrizio Grandoni, and Dieter Kratsch. 2009. A measure & conquer approach for the analysis of exact algorithms. *J. ACM* 56, 5 (2009). DOI:http://dx.doi.org/10.1145/1552285.1552286

Fedor V. Fomin, Dieter Kratsch, and Gerhard J. Woeginger. 2004. Exact (exponential) algorithms for the dominating set problem. In *Proceedings of the 30th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2004*. 245–256. DOI:http://dx.doi.org/10.1007/978-3-540-30559-0_21

Michael Held and Richard M. Karp. 1962. A dynamic programming approach to sequencing problems. *J. Soc. Indust. Appl. Math.* 10, 1 (1962), 196–210. DOI:http://dx.doi.org/10.1145/800029.808532

Russell Impagliazzo and Ramamohan Paturi. 2001. On the complexity of k-SAT. *J. Comput. System Sci.* 62, 2 (2001), 367–375. DOI:http://dx.doi.org/10.1006/jcss.2000.1727

Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. 2001. Which problems have strongly exponential complexity? *J. Comput. System Sci.* 63, 4 (2001), 512–530. DOI:http://dx.doi.org/10.1006/jcss.2001.1774

Joachim Kneis, Alexander Langer, and Peter Rossmanith. 2009. A fine-grained analysis of a simple independent set algorithm. In *Proceedings of the IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS2009*. 287–298. DOI:http://dx.doi.org/10.4230/LIPIcs.FSTTCS.2009.2326

Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. 2011. Slightly superexponential parameterized problems. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011*. 760–776. http://www.siam.org/proceedings/soda/2011/SODA11_059_lokshtanovd.pdf

Dániel Marx. 2007. On the optimality of planar and geometric approximation schemes. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2007*. 338–348. DOI:http://dx.doi.org/10.1109/FOCS.2007.50

Jesper Nederlof. 2009. Fast polynomial-space algorithms using Möbius inversion: Improving on Steiner tree and related problems. In *Proceedings of the 36th Internationcal Colloquium on Automata, Languages and Programming, ICALP 2009*. 713–725. DOI:http://dx.doi.org/10.1007/978-3-642-02927-1_59

Jesper Nederlof and Johan M. M. van Rooij. 2010. Inclusion/exclusion branching for partial dominating set and set splitting. In *Proceedings of the 5th International Symposium on Parameterized and Exact Computation, IPEC 2010*. 204–215. DOI:http://dx.doi.org/10.1007/978-3-642-17493-3_20

J. M. Robson. 1986. Algorithms for maximum independent sets. *Journal of Algorithms* 7, 3 (1986), 425–440. DOI:http://dx.doi.org/10.1016/0196-6774(86)90032-5

Rahul Santhanam and Srikanth Srinivasan. 2011. *On the limits of sparsification*. Tech report TR11-131. Electronic Colloquium on Computational Complexity (ECCC). http://eccc.hpi-web.de/eccc-reports/2011/TR11-131/

Rainer Schuler. 2005. An algorithm for the satisfiability problem of formulas in conjunctive normal form. *Journal of Algorithms* 54, 1 (2005), 40–44. DOI:http://dx.doi.org/10.1016/j.jalgor.2004.04.012

Patrick Traxler. 2008. The time complexity of constraint satisfaction. In *Proceedings of the 3rd International Workshop on Parameterized and Exact Computation, IWPEC 2008*. 190–201. DOI:http://dx.doi.org/10.1007/978-3-540-79723-4_18

Leslie G. Valiant. 1977. Graph-theoretic arguments in low-level complexity. In *Proceedings of the 6th Symposium on Mathematical Foundations of Computer Science, MFCS 1977*. 162–176. DOI:http://dx.doi.org/10.1007/3-540-08353-7_135

Johan M. M. van Rooij. 2011. *Exact exponential-time algorithms for domination problems in graphs*. Ph.D. Dissertation. Utrecht University.

Johan M. M. van Rooij, Jesper Nederlof, and Thomas C. van Dijk. 2009. Inclusion/exclusion meets measure and conquer. In *Proceedings of the 17th Annual European Symposium on Algorithms, ESA 2009*. 554–565. DOI:http://dx.doi.org/10.1007/978-3-642-04128-0_50

Emanuele Viola. 2009. On the power of small-depth computation. *Foundations and Trends in Theoretical Computer Science* 5, 1 (2009), 1–72. DOI:http://dx.doi.org/10.1561/0400000033

Ryan Williams. 2011. Non-uniform ACC circuit lower bounds. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC 2011*. 115–125. DOI:http://dx.doi.org/10.1109/CCC.2011.36